# Colorisation of Grayscale images using Conditional Diffusion

Adhvik Ramesh

May 14, 2025

## 1 Introduction

Generative models have seen significant advancements in recent years, with In recent years theres been rapid growth in the development and use of generative models in multiple use cases. Newer approaches like Generative Adversarial Networks (GANs) and diffusion models have shown promising results in the field of generating high-quality synthetic data. CGANs modify the original GAN architecture by introducing auxiliary information—such as class labels or attributes—into both the generator and discriminator. This modification enables more precise and controllable data generation, which is especially useful for tasks like image-to-image translation, style transfer, and generating images based on class information.

However, diffusion models are inspired by physical diffusion processes. These models learn to reconstruct data by reversing a sequence of noise additions, starting from random noise and gradually denoising it to produce a sample. This approach has recently achieved outstanding results in image generation making it relevant for our use case.

Our project investigates the design, implementation and comparative evaluation of CGANs and diffusion models for generative task of coloring greyscale images. We aim is to analyse and document the differences in the model performances for the same dataset (CIFAR-10)

### 1.1 Dataset Preparation

A custom PyTorch dataset class, `CIFAR10grayColor`, was developed to load color images and generate their greyscale counterparts using the `greyscale` transform. All images were resized to $128 \times 128$ pixels and converted to tensors.

# 2 Diffusion Model Training on CIFAR-10

We implemented a diffusion model on the CIFAR-10 dataset, using greyscale images as the conditional input. We trained the model to reconstruct the original color images from their greyscale versions, employing a denoising diffusion process.

## 2.1 Conditional GANs and Diffusion Models

**Conditional GANs (CGANs):** In CGANs, the generator and the discriminator are provided with the greyscale image as the condition. This conditioning guides the generator to produce colored images as per the specified context. This ensures that the model doesn't stray too far from the intended final product, making it more precise and reliable. **Diffusion Models:** Diffusion models are generative models that synthesize data by learning to reverse a noise-adding process. During training, noise is progressively added to data samples. The model then learns to de-noise these samples in steps, ultimately reconstructing high quality data from pure noise. This method has led to state of the art results in image synthesis, with improved stability and sample diversity compared to GANs.

## 2.2 Loss Function

The loss function was a weighted sum of three components:

- Mean Squared Error (MSE) for pixel-wise accuracy,
- Learned Perceptual Image Patch Similarity (LPIPS) for perceptual similarity,
- DeltaE in LAB color space for color fidelity.

This composite loss ensures that the generated images are visually, perceptually, and colorimetrically close to the ground truth.

## 2.3 Model Architectures

### Conditional GAN Framework

The system implements a dual-network adversarial architecture consisting of:
**Generator Network:**

- *Encoder-Decoder Structure:* Symmetrical architecture with 4× spatial downsampling (64→128 channels) followed by 4× upsampling through residual blocks
- *Residual Upblocks:* Combine nearest-neighbor interpolation with convolutional layers (128→64→32 channels) for detail-preserving upsampling

- *Output Layer:* 3-channel convolutional output with Tanh activation for [-1,1] normalized RGB predictions

**Discriminator Network:**

- *PatchGAN Architecture:* Processes 4-channel inputs (1 grayscale + 3 color) through 3 convolutional layers (64→128→256 channels)
- *Spectral Normalization:* Uses BatchNorm in intermediate layers with LeakyReLU (alpha=0.2) activations
- *Decision Layer:* Flattened 256×4×4 features fed to sigmoid-activated classifier for real/fake discrimination

### Diffusion Model Architecture

The denoising framework employs a time-conditioned UNet with several key innovations:

**Time-Aware UNet:**

- *Embedding Projection:* Sinusoidal time encodings projected through SiLU-activated linear layers (64→256D)
- *Residual Blocks:* GroupNorm-SiLU-Conv blocks with time embedding injection via learned linear projections
- *Multi-Scale Processing:* Downsampling path with 2× max pooling, bottleneck layer at 8×8 resolution, and skip-connected upsampling path

**Noise Scheduling:**

- *Cosine Schedule:* Implements improved noise variance schedule using cosine transitions over 1000 timesteps
- *Stable Clamping:* Ensures valid pixel ranges during reverse diffusion through output clamping

**Training Framework:**

- *Dual Objective:* Parallel training of noise-prediction (epsilon) and direct RGB-prediction (x0) UNets
- *Hybrid Loss:* Combines MSE (pixel accuracy), LPIPS (perceptual quality), and DeltaE2000 (color fidelity)
- *Optimization:* AdamW optimizer with gradient accumulation (4 steps) and mixed-precision training

The architecture enables both single-step GAN-based colorization and iterative diffusion-based refinement, providing complementary approaches for high-fidelity image synthesis.

### 2.4 Training Strategy

Training was conducted with mixed precision using PyTorch's Automatic Mixed Precision (AMP) to optimize memory usage. Gradient accumulation allowed for effective large-batch training on limited hardware. A cosine noise scheduler was used for the noise process, and the model was trained to predict the noise at each step.

### 2.5 Evaluation Metrics

The following metrics were used to eval the performance of the model:ce of the model:

- Peak Signal-to-Noise Ratio (PSNR)
- Structural Similarity Index Measure (SSIM)
- LPIPS

These metrics compare the denoised outputs with the original color images.

### 2.6 Results and Observations

Throughout training, sample outputs were recorded and validation was performed at the end of each epoch. The best-performing model, determined by the lowest validation loss, was saved for further analysis.

This approach balanced compitational efficiency with high-quality image synthesis, establishing a strong baseline for subsequent CGAN experiments.

## 3 Comparative Study: CGAN versus diffusion

To thoroughly compare generative architectures, we implemented and trained two models:

- **Standard CGAN**: A baseline conditional GAN using dense layers to generate images from greyscale inputs.
- **Diffusion Model**: A conditional diffusion model that reconstructs color images from greyscale inputs, using a UNet backbone for multi-scale feature extraction.

### 3.1 Training Setup

Both models were trained on greyscale-conditioned CIFAR-10 images resized to $128 \times 128$. Each model was optimized with hyperparameters like learning rate, optimizer, and loss function.

## 3.2 Evaluation Criteria

The models were evaluated using the following:

- PSNR and SSIM for measuring signal quality and structural accuracy,
- LPIPS for perceptual similarity,
- Inference time and training stability for practical deployment considerations.

## 3.3 Observations

- The CGAN produced plausible images but often lacked fine detail.

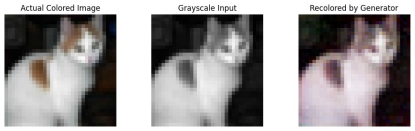


Figure 1: Example output from the standard CGAN.

- The diffusion model's performance was evaluated using two variants: one conditioned on noise and the other on RGB hints. While both models share the same UNet backbone, their loss functions differ: the noise model uses MSELoss against true noise, whereas the RGB model is trained using a combined loss function (LPIPS, MSE, and $\Delta E$) between the denoised output and the ground truth.
  - **RGB-conditioned model**:
    - **FID:** 2.95 (lower is better)
    - **KID:** $-6.27 \times 10^{-5}$ (lower is better)
    - **LPIPS:** 0.1056 (lower is better)
    - **PSNR:** 23.59 (higher is better)
    - **SSIM:** 0.6915 (close to 1 is better)
    - $\Delta E$**:** 16.65 (lower is better)
  - **Noise-conditioned model**:
    - **FID:** 90.11
    - **KID:** 0.0814
    - **LPIPS:** 0.4101
    - **PSNR:** 25.42
    - **SSIM:** 0.6968
    - $\Delta E$**:** 13.17

The RGB model excels in perceptual similarity (low FID, KID, and LPIPS), making it more visually realistic, while the noise model demonstrates superior structural and pixel-level fidelity, evident in its higher PSNR, SSIM, and lower $\Delta E$.


(a) Noise-prediction model output using MSE-focused loss


(b) RGB-prediction model output with perceptual (LPIPS + DeltaE) losses

Figure 2: Comparative outputs from our dual diffusion training framework. (a) The noise-conditional model preserves structural details through direct noise estimation. (b) The RGB-prediction model emphasizes color accuracy and perceptual quality through hybrid loss optimization.

# 4 Conclusion

In this project we have explored and compared two powerful generative modeling approaches—Conditional GANs and diffusion models—for the task of coloring greyscale images. We see that while CGANs are simpler and faster to train, they often lack fine detail. However, diffusion models, particularly the RGB-conditioned variant, produce more visually convincing results at the cost of higher computational demand and much longer training time.